

Programmation d'un PIC par Flowcode

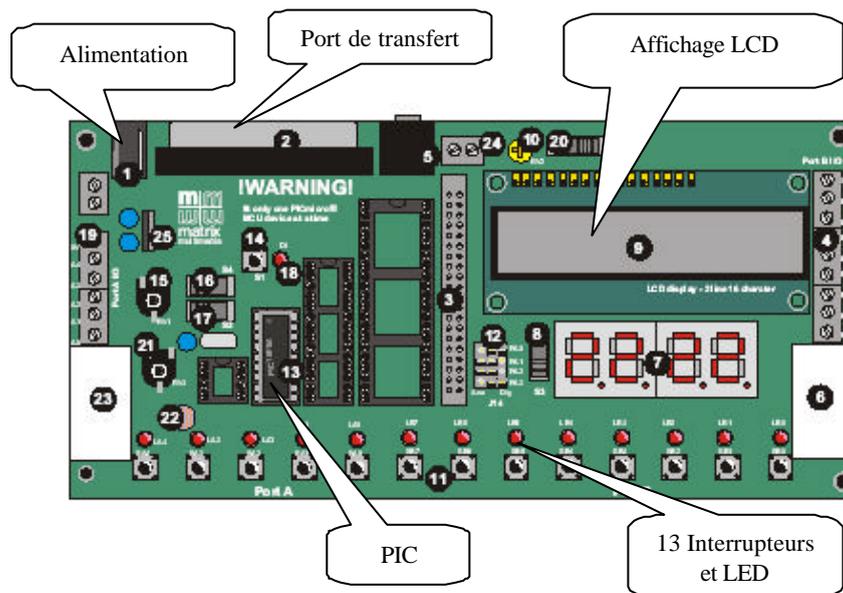
1 Présentation

Ce TP permet d'aborder la programmation d'un microcontrôleur de type PIC¹ par l'intermédiaire d'un logiciel algorithmique.

Un microcontrôleur est un microprocesseur simplifié auquel est associé dans le même boîtier l'ensemble des périphériques tels que les mémoires RAM et ROM, les entrées sorties et un *timer*. Certains disposent également de convertisseurs AN et NA.

La programmation d'un composant de ce type est généralement réalisée soit en assembleur soit dans un langage plus évolué tel que le C. Dans tous les cas, le programme est établi à l'aide d'un PC, il peut ensuite être simulé. Après l'élaboration, le programme est transféré en hexadécimal par l'intermédiaire d'un câble vers le microcontrôleur en utilisant un programmeur.

Le produit sur lequel nous allons travailler est constitué d'une carte d'évaluation supportant le microcontrôleur ainsi que des composants de test tels que des interrupteurs, des LED et des afficheurs.



Les softs sont composés d'un empilement qui permettent d'établir le programme sous la forme algorithmique. Pour transférer vers le PIC, le programme est tout d'abord converti en C puis en assembleur. Après cela il est compilé en hexadécimal en un fichier de type *.hex directement envoyé vers le composant.

L'intérêt du programme Flowcode est sa simplification car il ne nécessite pas une étude détaillée de la structure du composant utilisé. En effet, un programme rédigé en C ou en assembleur impose une connaissance parfaite de l'architecture² du microcontrôleur. Le

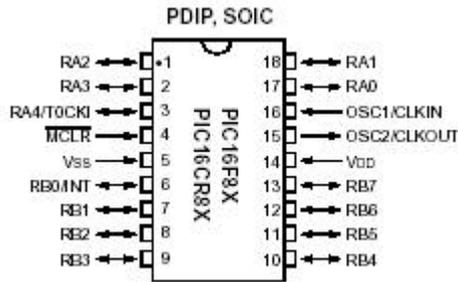
¹ PIC : *Programmable Integrated Circuit*

² La documentation technique du PIC est constituée de 184 pages en anglais.

TP réalisé en 6 heures

logiciel Flowcode se charge de transcoder l’algorithme quel que soit le type de composant utilisé.

Le composant avec lequel nous travaillons est un microcontrôleur classique de référence PIC 16F84 A.



Il est constitué de 18 broches dont 13 correspondent à deux ports A et B.

Rappelons qu’un port permet de faire transiter une donnée binaire exprimée par 0 ou 1. Le port A est composé de 5 bits RA0/4 et le port B de 8 bits RB0/7.

Chacune de ces 13 «pattes » peut être configurée soit en entrée soit en sortie.

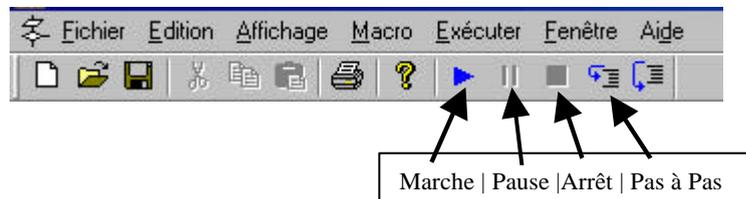
Le travail qui vous est demandé est composé de deux parties :

- L’analyse d’un programme rédigé avec *Flowcode*
- La conception d’un programme sous *Flowcode*.

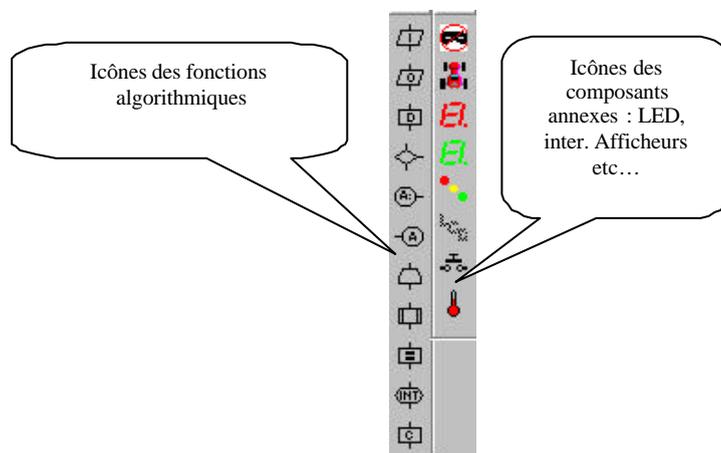
2 Présentation succincte de *Flowcode*

La documentation technique du logiciel permet d’aborder très rapidement son utilisation. Néanmoins, quelques éléments peuvent résumer son fonctionnement.

A la barre de menu classique de Windows est ajoutée cinq icônes qui permettent de simuler le programme.



La barre de menu vertical regroupe à gauche les 10 fonctions basiques de création de l’algorithme. A droite les icônes correspondent aux composants périphériques au PIC tels que les afficheurs, *Led* et interrupteurs.



-  [Propriétés de l'icône Entrée](#)
-  [Propriétés de l'icône Sortie](#)
-  [Propriétés de l'icône Pause](#)
-  [Propriétés de l'icône Décision](#)
- 
-  [Propriétés de l'icône Point de jonction](#)
-  [Propriétés de l'icône Boucle](#)
-  [Propriétés de l'icône Macro](#)
-  [Propriétés de l'icône Calculs](#)
-  [Propriétés de l'icône Interruption](#)
-  [Propriétés de l'icône Code C](#)

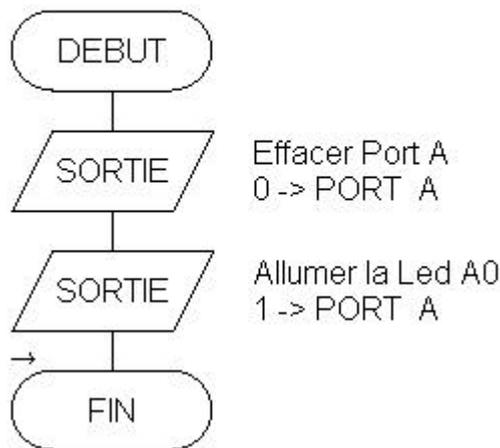
Il sera demandé d'étudier les programmes étape par étape et de rédiger un compte rendu.

3 Travail d'analyse

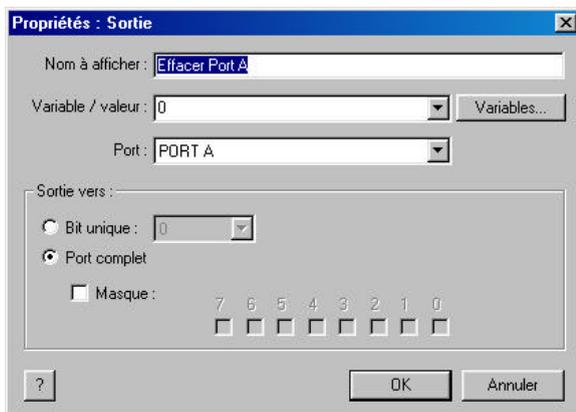
Exercice TD_01.fcf

Dans le répertoire *Travaux dirigés* ouvrir le fichier TD_01.fcf. Le schéma suivant représente l'algorithme réalisé sous *Flowcode*. Ce programme est borné par les deux blocs DEBUT et FIN, il fait appel à deux fonctions SORTIE.

Allumer une LED (A0)



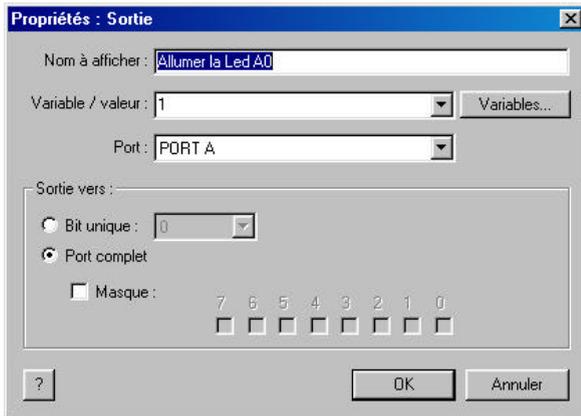
Avec la souris pointer le curseur sur la première fonction *SORTIE* puis activer le curseur droit et valider la ligne *Propriété*. La fenêtre suivante s'ouvre :



SORTIE : effacer port A :

Par cette fonction, les cinq bits du port A sont mis à zéro. (valeur = 0)

Comme les broches du port A sont reliées à cinq LED(s), ces cinq LED(s) correspondantes sont éteintes.



SORTIE : Allumer les LEDs A :

Le port A a ici pour valeur **1**. Cette valeur est décimale. Mais comme $1_{(10)} = 001_{(2)}$, seul la LED repérée A0 va s'allumer.

Si la valeur de A était $4_{(10)}$, les LED(s) seraient allumées en correspondances avec la valeur binaire suivante : $0100_{(2)}$.

Analyser et étudier les informations proposées dans l'aide Flowcode.

3.1 Travail

Pour tester le fonctionnement du programme, activer l'icône MARCHE, les LEDs de l'écran doivent s'allumer.

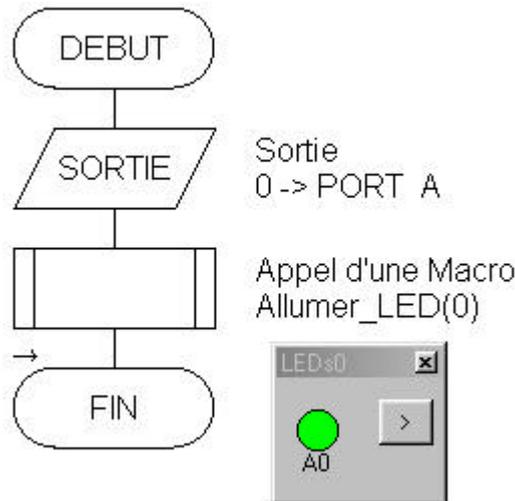
Après cela, actionner l'icône STOP. Puis reprendre l'exécution en mode Pas à Pas.

3.2 Transfert vers le PIC

Avec la souris valider Exécuter puis transfert vers le PIC. Si le lien avec la carte est correct, le transfert doit être validé et les quatre LEDs de la carte doivent s'allumer.

3.3 Variante de l'exercice précédent

La sortie qui permet d'allumer la LED est remplacée par une Macro qui réalise la même fonction.



4 Exercice suivant

Le but de cet exercice est de réaliser un programme qui allume le LED A0 uniquement si la variable TOTO a pour valeur 1 en base 10. Cette variable TOTO est issue de la lecture au clavier de l'interrupteur A0, cet interrupteur est simulé par la touche 0 du pavé numérique.

L'algorithme est représenté par le schéma suivant.

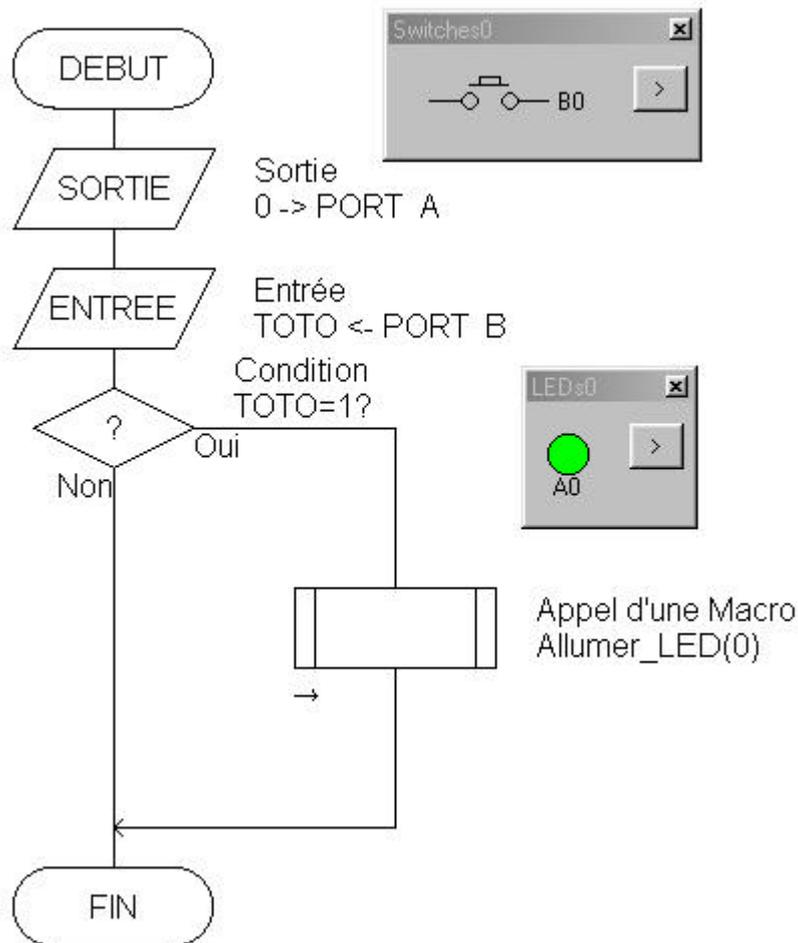
4.1 Travail

En utilisant la documentation FLOWCODE :

- Reproduire cet algorithme sous Flowcode.
- Créer la variable **TOTO**.
- Associer les blocs **interrupteurs** et **LED**, vérifier les **connexions** et les **propriétés** de ces deux composants en respect de l'énoncée.
- Associez à l'interrupteur le clavier numérique du PC avec les fonctions SWITCH.

Vérifier la simulation avec le mode **Pas à PAS**

Transférer vers la carte et vérifier le fonctionnement



4.2 Ajout d'actions

Retourner à l'écran vers Flowcode. Il est maintenant demandé d'ajouter une fonction temporisation pour que :

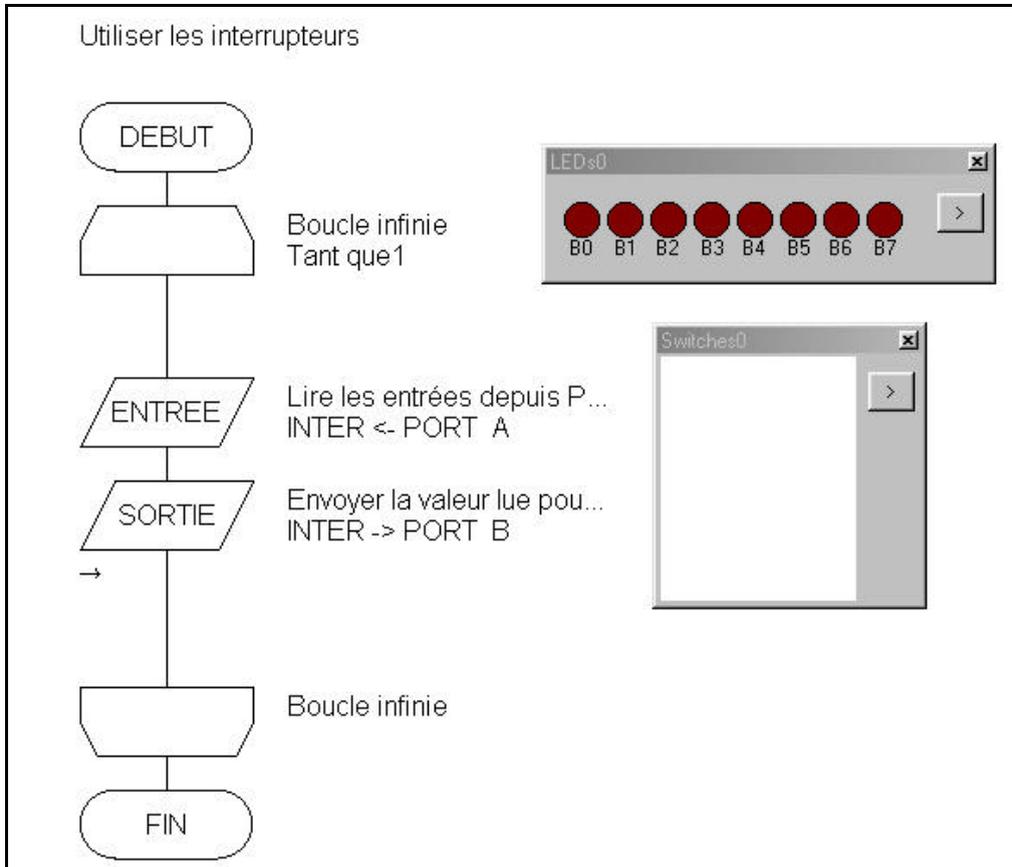
La Led reste allumée uniquement pendant 1 seconde.

Reprendre la documentation Flowcode concernant l'icône temporisation et composez ensuite l'algorithme.

Simuler à l'écran. Si la simulation fonctionne correctement, transférer le programme vers le PIC et tester sur la carte.

5 Analyse d'un algorithme

A droite du schéma, nous voyons les deux fonctions qui sont utilisés dans le programme et qui permettent de vérifier la simulation : un bloc de 8 LED et un bloc de quatre boutons poussoirs. (*qui seront simulés sur le PC par les touches 0 à 3 du clavier*)



5.1 Travail n°1

En utilisant la documentation de Flowcode, il est demandé de rechercher et d'étudier chacun des 3 symboles/fonctions utilisées dans ce programme.

A partir de cette analyse : que réalise ce programme ?

5.2 Travail n°2

Ouvrir le logiciel Flowcode et construire ce programme en utilisant les icônes correspondantes. Compléter ce programme en activant la *propriété* de chaque icône.

5.3 Travail n°2

Activer la simulation de ce programme en rendant actif le bloc interrupteur par le clavier du PC pour cela il est indispensable d'étudier la doc Flowcode.

5.4 Travail n°3

Si la simulation à aboutie, transférer le programme vers la carte.

Si ce transfert à abouti, vérifier directement sur la carte que le programme fonctionne en activant les boutons poussoirs correspondant.

5.5 *Création d'un programme*

Il est demandé de créer l'algorithme à partir des points suivants :

- Initialisez le port B à 0, pour éteindre toutes les LEDs du port B.
- Par action sur le bouton poussoir A0, la LED B0 doit s'allumer.
- Le bouton poussoir A0 doit être simulé par la touche [0] du clavier numérique de l'ordinateur.
- Par l'action sur le bouton poussoir A1, la LED doit s'éteindre. (simulé par la touche 1 du clavier)
- Une action sur A1 si la LED est éteinte n'a pas d'effet.
- Une action sur A0 si la LED est allumée n'a pas d'effet.
- Ces fonctions doivent être répétitives à l'infini.

Tester la simulation à l'écran. Si cette simulation fonctionne transférer le programme vers la carte d'évaluation.

Proposition de corrigé de l'exercice

